# ForestFormer3D: A Unified Framework for End-to-End Segmentation of Forest LiDAR 3D Point Clouds

Supplementary Material

## 1. Overview

In this supplementary material, we provide:
- implementation details (Sec. 2),
- additional dataset details and point cloud annotation (Sec. 3),
- a quantitative comparison including both individual tree segmentation and semantic segmentation across nine regions in the FOR-instanceV2 test split (Sec. 4),
- additional ablation studies (Sec. 5),
- qualitative results (Sec. 6), and
- relation to forest domain specific metrics (Sec. 7).

## 2. Implementation details

**3D sparse U-Net architecture.** The 3D sparse U-Net used for sparse tensor feature extraction is shown in Fig. 1.

**Loss calculation.** All instance masks, predicted by transformer decoder, are supervised by losses including binary cross-entropy (BCE), Dice, and score loss.

$$\mathcal{L}_{\text{bce}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log \sigma(\hat{y}_i) + (1-y_i)\log(1-\sigma(\hat{y}_i))\right), \tag{1}$$

$$\mathcal{L}_{\text{dice}} = 1 - \frac{2\sum_{i=1}^{N}\sigma(\hat{y}_i)y_i + 1}{\sum_{i=1}^{N}\sigma(\hat{y}_i) + \sum_{i=1}^{N} y_i + 1}, \tag{2}$$

where $N$ is the number of voxels, $\hat{y}_i$ is the predicted mask logits at voxel $i$, $y_i \in \{0,1\}$ is the GT mask value, and $\sigma(\cdot)$ represents the sigmoid function. The additive constant $+1$ in Eq. (2) prevents division by zero.

The score loss supervises the predicted confidence scores of all predicted instance masks using mean squared error:

$$\mathcal{L}_{\text{score}} = \frac{1}{M}\sum_{j=1}^{M}(\hat{s}_j - s_j)^2, \tag{3}$$

where $M$ is the number of predicted masks, $\hat{s}_j$ is the predicted score for predicted mask $j$, and $s_j$ is set as the IoU between the predicted mask and its best matched GT mask, or zero if no GT overlaps. IoU is computed based on the number of intersecting voxels divided by the union of the predicted and GT mask voxels.

To facilitate query point selection, we first construct a 5D embedding space where voxels from the same instance are close while those from different instances are apart. This embedding is learned using discriminative loss:

$$\mathcal{L}_{\text{disc}} = \mathcal{L}_{\text{var}} + \mathcal{L}_{\text{dist}} + 0.001 \cdot \mathcal{L}_{\text{reg}}. \tag{4}$$

$\mathcal{L}_{\text{var}}$ encourages intra-instance compactness by pulling voxel embeddings within the same instance toward their mean. $\mathcal{L}_{\text{dist}}$ enforces inter-instance separation, and $\mathcal{L}_{\text{reg}}$ constrains instance centroids to stay near the origin. Let $C$ denote the total number of tree instances, and $\mathcal{I}_c$ the set of voxels belonging to instance $c$. Each voxel $i$ has an embedding $f_i$, and the mean embedding of instance $c$ is $\mu_c$. We define $N_c$ as the number of voxels in instance $c$. The margins $\delta_v$ and $\delta_d$ control the desired intra-instance compactness and inter-instance separation, respectively. Three loss terms are calculated by:

$$\mathcal{L}_{\text{var}} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i\in\mathcal{I}_c}\left[\max\left(\|f_i - \mu_c\|_1 - \delta_v, 0\right)\right]^2, \tag{5}$$

$$\mathcal{L}_{\text{dist}} = \frac{1}{C(C-1)}\sum_{c_1 \neq c_2}\left[\max\left(2\delta_d - \|\mu_{c_1} - \mu_{c_2}\|_1, 0\right)\right]^2, \tag{6}$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{C}\sum_{c=1}^{C}\|\mu_c\|_1. \tag{7}$$

We set $\delta_v = 0.5$ and $\delta_d = 1.5$ following prior work [4, 9, 11].

In addition, to ensure that query points are sampled from tree voxels, we train a tree and non-tree classification head to distinguish tree and non-tree voxels using the BCE loss:

$$\mathcal{L}_{\text{binary}} = -\frac{1}{N}\sum_{i=1}^{N}(l_i \log \hat{l}_i + (1-l_i)\log(1-\hat{l}_i)), \tag{8}$$

where $l_i$ is the GT label for voxel $i$, and $\hat{l}_i$ is the predicted class probability.

**Other implementation details.** Point clouds are voxelized to a resolution of 0.2 m. Data augmentation includes random horizontal flipping, random rotation around the Z-axis, and random scaling by a factor between 0.8 and 1.2. We train our model on a single NVIDIA A100 GPU with 80 GB memory, using a batch size of 2 for 3000 epochs.

To stabilize training, we first pretrain two MLP heads introduced in ISA-guided query point selection for 1000 epochs before jointly training the full model. Specifically, the 5D embedding feature head is trained with the discriminative loss (see Eq. (4)) to enforce compact intra-instance representations and inter-instance separation, while the binary classification head is trained with a binary cross-entropy loss to distinguish tree voxels from non-tree voxels (see Eq. (8)). During this warm-up stage, the decoder remains frozen, ensuring that the MLP heads produce stable embeddings and classifications before integrating with the full network. After this stage, we jointly train all components, including the 3D sparse U-Net and the transformer decoder, optimizing all loss terms together.

For inference, we sample 300 query points within each cylindrical region of radius 16 m with a stride of 4 m using a sliding window approach. The best score threshold on mask removal is 0.4 according to our ablation studies (Fig. 3(d)).

## 3. Dataset details and point cloud annotation

Tab. 1 summarizes the key characteristics of the FOR-instanceV2 dataset and the additional test datasets.

For the BlueCat data annotation, all trees were manually labeled using 3D Forest software [8] by two independent annotators working on separate, non-overlapping datasets. After labeling, the trees were matched to field measurements based on spatial proximity. This pairing was validated by comparing tree heights derived from the point cloud with those estimated from DBH-based allometry. In addition, every tree with a DBH of 1 cm or greater was field-verified to eliminate false positives, and any obvious labeling errors were manually refined.

For the NIBIO2 data, annotations were performed using CloudCompare software (https://www.cloudcompare.org, V 2.12.4) by two annotators, and the results were later reviewed to ensure consistency and accuracy.

## 4. Comparison to baselines across nine regions in the FOR-instanceV2 test split

We evaluate ForestFormer3D's quantitative performance across nine different forest regions in the FOR-instanceV2 test set and compare it against other methods (see Tab. 2). It is important to note that CHM-based YOLOv5 [7], ForAINet [12], and SegmentAnyTree [10] are trained on the original FOR-instance dataset [6], which does not include data from the NIBIO_MLS, BlueCat, and Yuchen regions. As shown in Tab. 2, ForestFormer3D consistently achieves the best results for individual tree segmentation across all nine regions except for SCION. In simpler forest areas, such as CULS, NIBIO, SCION, and NIBIO_MLS, ForestFormer3D maintains high F1 and Cov scores. In more challenging environments, such as TUWIEN (where trees are closely intertwined), RMIT and Yuchen (with sparse point densities), and NIBIO2 and BlueCat (characterized by both large canopy trees and closely packed understory trees), ForestFormer3D surpasses all prior baselines. It demonstrates substantial improvements in regions that were previously difficult for existing methods. For example, in the RMIT region, ForestFormer3D improves the F1 by 6.8 pp over the second best one. In the SCION region, the Cov metric shows an increase of 5.1 pp. The NIBIO2 region is particularly challenging due to the presence of both understory and canopy trees, as noted in prior work [12]. Despite this, ForestFormer3D outperforms the OneFormer3D baseline by 8.2 pp in F1 and achieves a 17.6 pp improvement over the reported F1 of 72.8% from ForAINet [12].

For semantic segmentation, ForestFormer3D demonstrates competitive or superior performance in most regions, with slight decreases in mIoU observed only in the CULS and NIBIO_MLS regions compared to the best-performing method.

## 5. Additional ablation studies

**Input radius size and number of query points.** Fig. 2 compares the effect of different cylinder input radii and the number of query points on individual tree segmentation metrics (F1 and Cov) within the FOR-instanceV2 test split. Fig. 2(a) shows the performance of OneFormer3D, while Fig. 2(b) shows the performance of ForestFormer3D. `r8_qp200` represents a cylinder radius of 8 m with 200 query points, with similar interpretations for other settings. For ForestFormer3D, increasing the input radius and the number of query points generally results in higher F1 scores. However, due to GPU memory limitations, further increases in these parameters were not explored. For OneFormer3D, achieving an optimal balance between the input radius and the number of query points is crucial to better performance.

**Score threshold on mask removal.** We evaluate the effect of the score threshold on Prec, Rec, F1, and Cov using the FOR-instanceV2 dataset (see Fig. 3). During inference, masks with scores below the threshold are removed as unreliable, affecting the balance between retaining true positives and removing false positives. As shown in Fig. 3, higher thresholds improve Prec (Fig. 3(b)) by reducing false positives but decrease Cov (Fig. 3(a)) and Rec (Fig. 3(c)) by discarding true positives. Fig. 3(d) highlights the trade-off between Prec and Rec. A threshold of 0.4 achieves the best F1 score in both validation and test sets, making it the optimal choice for our framework.

**Point cloud density.** We evaluated the impact of different point cloud densities on segmentation performance. The original FOR-instanceV2 dataset was downsampled to seven densities: 10, 25, 50, 75, 100, 500, and 1000 points per square meter (pts/m$^2$). The results are shown in Fig. 4. As the point density decreases, all metrics experience varying degrees of degradation. In particular, mIoU and Prec remain relatively stable in different densities, while Rec, Cov,

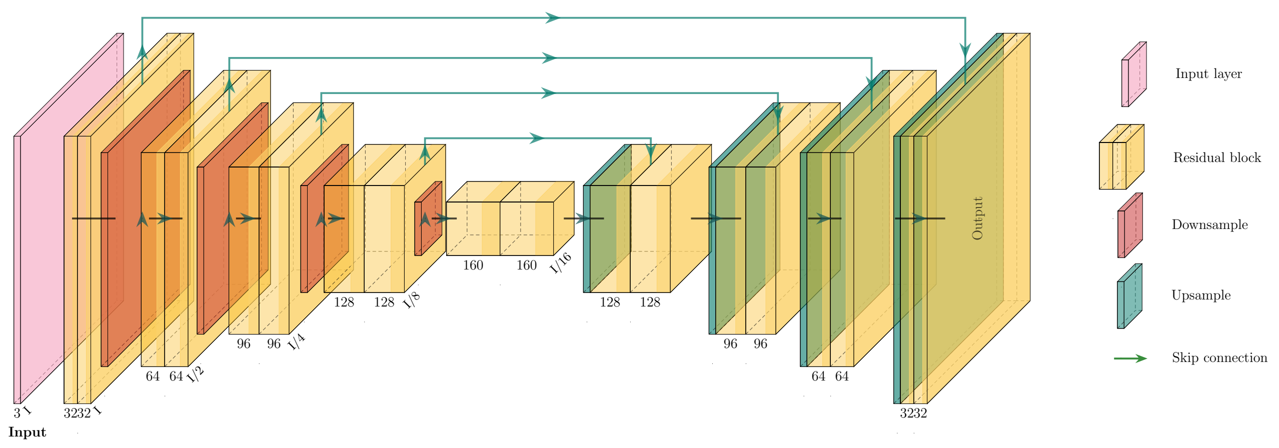

Figure 1. The detailed structure of the 3D sparse U-Net architecture used for feature extraction from the voxelized 3D point cloud.

Table 1. Characteristics of the FOR-instanceV2 dataset and additional test datasets in different geographic regions. The FOR-instanceV2 dataset contains unique individual tree IDs for each point, as well as semantic labels for ground, wood, and leaf. The additional test data, on the other hand, includes only tree and non-tree labels, along with individual tree IDs for each point.

| | Region name | Forest type (Tree species) | Scanning mode (Sensor) | Number of plots | | | Number of trees | Country |
|---|---|---|---|---|---|---|---|---|
| | | | | Train | Val | Test | | |
| FOR-instanceV2 | CULS [6] | Coniferous dominated temperate forest (*Pinus sylvestris*) | ULS (Riegl VUX-1) | 1 | 1 | 1 | 37 | Czech Republic |
| | NIBIO [6] | Coniferous dominated boreal forest (*Picea abies*, *Pinus sylvestris*, *Betula sp.* (few)) | ULS (Riegl MiniVUX-1) | 8 | 6 | 6 | 575 | Norway |
| | RMIT [6] | Native dry sclerophyll eucalypt forest (*Eucalyptus sp.*) | ULS (Riegl MiniVUX-1) | 1 | 0 | 1 | 223 | Australia |
| | SCION [6] | Non-native pure coniferous temperate forest (*Pinus radiata*) | ULS (Riegl MiniVUX-1) | 2 | 1 | 2 | 135 | New Zealand |
| | TUWIEN [6] | Deciduous dominated temperate forest (Deciduous species) | ULS (Riegl VUX-1) | 1 | 0 | 1 | 150 | Austria |
| | NIBIO2 | Coniferous dominated boreal forest (*Pinus sylvestris*, *Picea abies*, *Betula sp.*) | ULS (Riegl VUX-1) | 29 | 6 | 15 | 3062 | Norway |
| | NIBIO_MLS [10] | Coniferous dominated boreal forest (*Picea abies*, *Pinus sylvestris*, *Betula sp.*) | MLS (GeoSLAM ZEB-HORIZON) | 4 | 1 | 1 | 258 | Norway |
| | BlueCat | Deciduous temperate forest (Mixed species) | TLS (Leica P20) | 1 | 1 | 1 | 6304 | Czech Republic |
| | Yuchen [2] | Tropical forest | ULS (Riegl MiniVUX-1) | 1 | 1 | 1 | 281 | French Guiana |
| Additional | Wytham woods [3] | Deciduous temperate forest (*Fraxinus excelsior*, *Acer pseudoplatanus*, *Corylus avellana*) | TLS (RIEGL VZ-400) | 0 | 0 | 1 | 835 | United Kingdom |
| | LAUTx [1] | Mixed temperate broad leaved, coniferous forest (Mixed species) | MLS (GeoSLAM ZEB-HORIZON) | 0 | 0 | 6 | 516 | Austria |

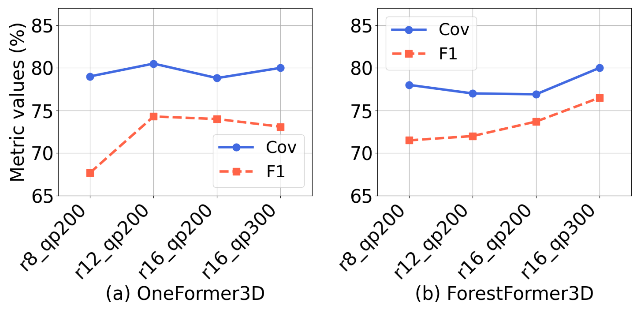

Figure 2. Impact of cylinder input radius and number of query points on individual tree segmentation metrics for FOR-instanceV2 test split.

and F1 show more noticeable drops when the density falls below 500 pts/m². This suggests that the method may face certain limitations in extremely sparse point cloud scenarios.

To improve robustness under varying point cloud densities, we retrained ForestFormer3D using the multi density augmentation strategy from SegmentAnyTree [10]. The training set combined point clouds from eight densities (i.e., the original resolution and seven downsampled versions). As shown in Fig. 5, this strategy improves individual tree segmentation under sparse conditions (e.g., $< 100$ pts/m²), yielding higher F1 scores. However, it brings no benefit for dense inputs and can even degrade performance. These results suggest that simple multi-density training is insufficient to address the challenges posed by point density variation, which remains an open problem.

Table 2. Comparison with other baselines on the individual tree segmentation and semantic segmentation for different forest regions in FOR-instanceV2 test split. The best results are in bold, and the second best ones are underlined.

| Region | Method | Individual Tree Seg. (%) | | | | Semantic Seg. (%) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Prec | Rec | F1 | Cov | Ground | Wood | Leaf | mIoU |
| CULS | CHM-based YOLOv5 [7] | **100.0** | **100.0** | **100.0** | – | – | – | – | – |
| | ForAINet [12] | 87.0 | **100.0** | 93.0 | 98.2 | – | – | – | – |
| | SegmentAnyTree [10] | **100.0** | **100.0** | **100.0** | – | – | – | – | – |
| | ForAINetV2_R16 | **100.0** | **100.0** | **100.0** | 96.6 | **99.8** | 67.2 | 96.4 | 87.8 |
| | OneFormer3D [5] | 95.0 | 95.0 | 95.0 | 94.6 | **99.8** | **69.0** | **96.5** | **88.5** |
| | ForestFormer3D (ours) | **100.0** | **100.0** | **100.0** | **99.4** | **99.8** | 66.1 | 96.2 | 87.4 |
| NIBIO | CHM-based YOLOv5 [7] | 87.0 | 72.0 | 79.0 | – | – | – | – | – |
| | ForAINet [12] | 96.4 | 88.4 | 92.4 | 79.4 | – | – | – | – |
| | SegmentAnyTree [10] | 91.0 | 88.0 | 89.5 | – | – | – | – | – |
| | ForAINetV2_R16 | **98.1** | 89.4 | 93.4 | 82.4 | 97.9 | 62.1 | 93.3 | 84.4 |
| | OneFormer3D [5] | 79.2 | **96.2** | 86.6 | **88.9** | 97.8 | 60.7 | 93.0 | 83.9 |
| | ForestFormer3D (ours) | 97.7 | 95.8 | **96.7** | 88.8 | **98.1** | **64.2** | **93.5** | **85.2** |
| RMIT | CHM-based YOLOv5 [7] | 70.0 | 62.0 | 65.0 | – | – | – | – | – |
| | ForAINet [12] | 75.9 | 64.1 | 69.5 | 60.6 | – | – | – | – |
| | SegmentAnyTree [10] | **83.0** | 69.0 | 75.4 | – | – | – | – | – |
| | ForAINetV2_R16 | 74.5 | 59.4 | 66.1 | 60.3 | **98.3** | 55.8 | 92.7 | 82.3 |
| | OneFormer3D [5] | 70.3 | 81.2 | 75.4 | **73.7** | 98.1 | 52.7 | **92.8** | 81.2 |
| | ForestFormer3D (ours) | 81.5 | **82.8** | **82.2** | 73.5 | 98.2 | **58.1** | 92.7 | **83.0** |
| SCION | CHM-based YOLOv5 [7] | 91.0 | 91.0 | 91.0 | – | – | – | – | – |
| | ForAINet [12] | 96.0 | 87.7 | 91.5 | 83.1 | – | – | – | – |
| | SegmentAnyTree [10] | 93.0 | 92.0 | 92.5 | – | – | – | – | – |
| | ForAINetV2_R16 | **100.0** | 90.4 | **95.0** | 83.2 | **99.7** | 61.9 | 95.1 | 85.6 |
| | OneFormer3D [5] | 58.7 | **92.4** | 71.7 | 83.6 | **99.7** | 61.7 | 95.0 | 85.5 |
| | ForestFormer3D (ours) | 97.1 | **92.4** | 94.7 | **88.7** | **99.7** | **64.4** | **95.5** | **86.5** |
| TUWIEN | CHM-based YOLOv5 [7] | 41.0 | 23.0 | 30.0 | – | – | – | – | – |
| | ForAINet [12] | 66.6 | **71.4** | 69.4 | **58.3** | – | – | – | – |
| | SegmentAnyTree [10] | 55.0 | 46.0 | 50.1 | – | – | – | – | – |
| | ForAINetV2_R16 | 68.2 | 42.9 | 52.6 | 47.9 | 98.2 | **53.3** | **94.5** | **82.0** |
| | OneFormer3D [5] | 41.5 | 62.9 | 50.0 | 54.8 | 98.4 | 48.4 | 94.2 | 80.3 |
| | ForestFormer3D (ours) | **92.0** | 65.7 | **76.7** | 54.8 | **98.5** | 52.8 | 94.2 | 81.8 |
| NIBIO2 | ForAINet [12] | 83.5 | 64.5 | 72.8 | – | – | – | – | |
| | ForAINetV2_R16 | 91.8 | 72.9 | 80.6 | 70.0 | 96.5 | 52.9 | 95.7 | 81.7 |
| | OneFormer3D [5] | 79.2 | 85.9 | 82.2 | 79.1 | **96.9** | 54.4 | 95.8 | 82.3 |
| | ForestFormer3D (ours) | **94.6** | **86.9** | **90.4** | **80.2** | **96.9** | **56.6** | **95.9** | **83.1** |
| NIBIO_MLS | ForAINetV2_R16 | 95.5 | 91.3 | 93.3 | 84.4 | 98.8 | 74.9 | 86.4 | 86.7 |
| | OneFormer3D [5] | 79.3 | **100.0** | 88.5 | **89.8** | **98.9** | 75.3 | **87.2** | **87.1** |
| | ForestFormer3D (ours) | **100.0** | 91.3 | **95.5** | 85.4 | 94.5 | **77.6** | 87.1 | 86.4 |
| BlueCat | ForAINetV2_R16 | 71.8 | 27.9 | 40.2 | 32.8 | – | **73.3** | **94.7** | **84.0** |
| | OneFormer3D [5] | 59.7 | 47.1 | 52.6 | 48.3 | – | 64.4 | 93.0 | 78.7 |
| | ForestFormer3D (ours) | **84.5** | **48.6** | **61.7** | **48.8** | – | 69.9 | 93.9 | 81.9 |
| Yuchen | ForAINetV2_R16 | 78.3 | 75.0 | 76.6 | 74.9 | 99.4 | **51.7** | **98.1** | **83.0** |
| | OneFormer3D [5] | 52.9 | 75.0 | 62.1 | 71.6 | 99.6 | 50.0 | 97.9 | 82.5 |
| | ForestFormer3D (ours) | **90.5** | **79.2** | **84.4** | **79.2** | **99.7** | 49.7 | **98.1** | 82.5 |

## 6. Qualitative results

To provide an intuitive understanding of how segmentation metrics relate to actual segmentation quality, we present visualizations of ForestFormer3D's results in individual tree segmentation and semantic segmentation tasks.

Fig. 6 shows both individual tree segmentation and semantic segmentation predictions, along with corresponding ground truth segmentations, across nine different forest regions in the FOR-instanceV2 test split. The results show that our ForestFormer3D performs well under various conditions, including data collected from different sensors, different point densities, varying forest types and geographical locations (*i.e.*, boreal, temperate and tropical forests), and complex environments where both large and small trees coexist. These visual results demonstrate the robustness of ForestFormer3D in handling diverse and challenging forest scenarios.

Fig. 7 displays individual tree segmentation results with predicted and ground truth segmentations for ForestFormer3D on the Wytham woods and LAUTx datasets. Fig. 8 presents three representative examples, progressing from simple forest scenes to densely packed environments with both large canopy trees and understory ones. For each example, we include visualizations of the ISA-guided query points, the predicted masks generated from these points, and the final individual tree segmentation predictions, along with the ground truth for comparison.

Fig. 9 presents a visual comparison of ForestFormer3D

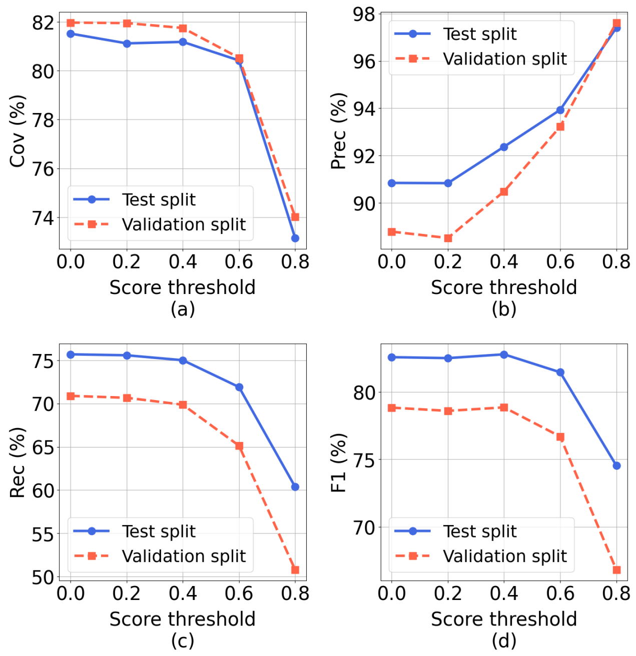

Figure 3. Effect of score threshold on key metrics of individual tree segmentation (Prec, Rec, F1 and Cov) for FOR-instanceV2 dataset.

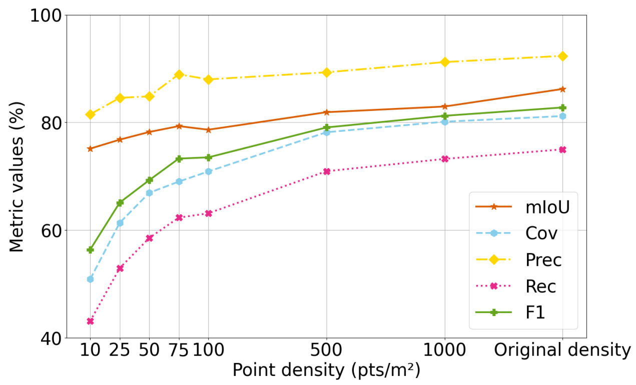

Figure 4. Effect of point density on metrics of individual tree segmentation (Prec, Rec, F1 and Cov) and semantic segmentation (mIoU) for FOR-instanceV2 test split.

and baseline methods. In Example 1 and Example 2, ForestFormer3D demonstrates its ability to effectively mitigate over-segmentation and under-segmentation compared to other methods. However, in Example 2, small trees are occasionally missed, indicating a potential limitation in detecting such cases. In Example 3, heavily inclined small trees near the ground pose significant challenges for all methods, leading to either missed detections, over-segmentation, or incorrect assignment to neighboring larger trees. Addressing these complex scenarios remains an open problem for future research.

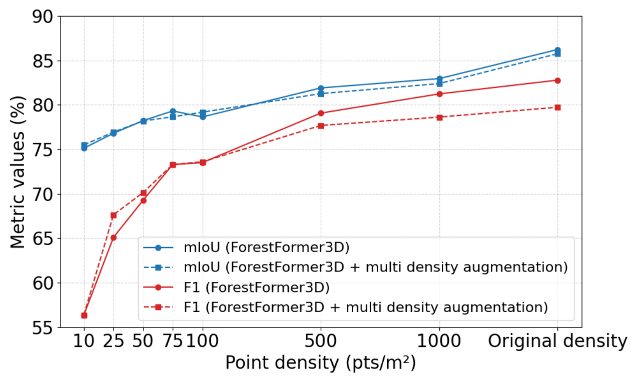

Figure 5. Effect of multi density augmentation on individual tree segmentation (F1) and semantic segmentation (mIoU) on the FOR-instanceV2 test split.

## 7. Relation to forest domain specific metrics

While domain-specific metrics have long been used in forestry research, we adopt Prec, Rec, and F1 to align with standard evaluation practices in computer vision. This allows for direct comparability across domains while maintaining consistency with traditional forestry metrics. Specifically, completeness corresponds to Rec, omission error to $1-\text{Rec}$, commission error to $1-\text{Prec}$, and F-score remains equivalent to F1. These relationships have been previously established in forestry research [12], ensuring that our reported results remain interpretable for both forestry and computer vision communities.

It is worth noting that SegmentAnyTree [10] reports locally computed F1 scores, whereas we adopt global metrics consistent with ForAINet [12]. This difference in evaluation metrics explains the discrepancy in reported F1 scores across methods.

## References

[1] Tockner Andreas, Gollob Christoph, Ritter Tim, and Nothdurft Arne. LAUTx - individual tree point clouds from austrian forest inventory plots, 2022. 3

[2] Yuchen Bai, Jean-Baptiste Durand, Grégoire Vincent, and Florence Forbes. Semantic segmentation of sparse irregular point clouds for leaf/wood discrimination. In *NIPS*, pages 48293–48313, 2023. 3

[3] Kim Calders, Hans Verbeeck, Andrew Burt, Niall Origo, Joanne Nightingale, Yadvinder Malhi, Phil Wilkes, Pasi Raumonen, Robert GH Bunce, and Mathias Disney. Laser scanning reveals potential underestimation of biomass carbon in temperate forest. *Ecological Solutions and Evidence*, 3(4): e12197, 2022. 3

[4] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation for autonomous driving. In *CVPRW*, pages 478–480, 2017. 1

[5] Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. OneFormer3D: One transformer for

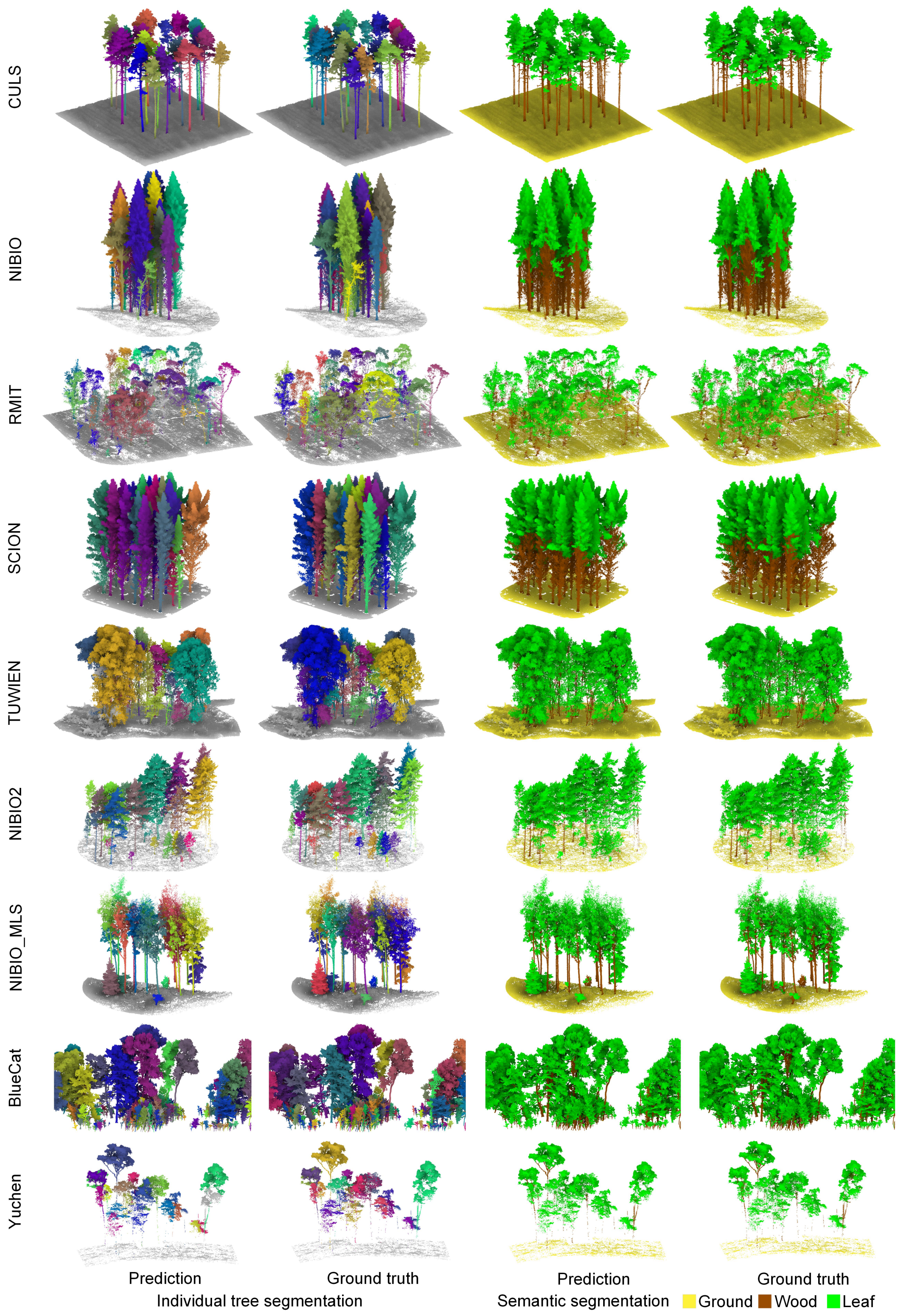

Figure 6. Visualization of ForestFormer3D segmentation results for nine different regions in FOR-instanceV2 test split. The colors for the individual trees are assigned randomly.

unified point cloud segmentation. In *CVPR*, pages 20943–20953, 2024. 4

[6] Stefano Puliti, Grant Pearse, Peter Surový, Luke Wallace, Markus Hollaus, Maciej Wielgosz, and Rasmus Astrup.

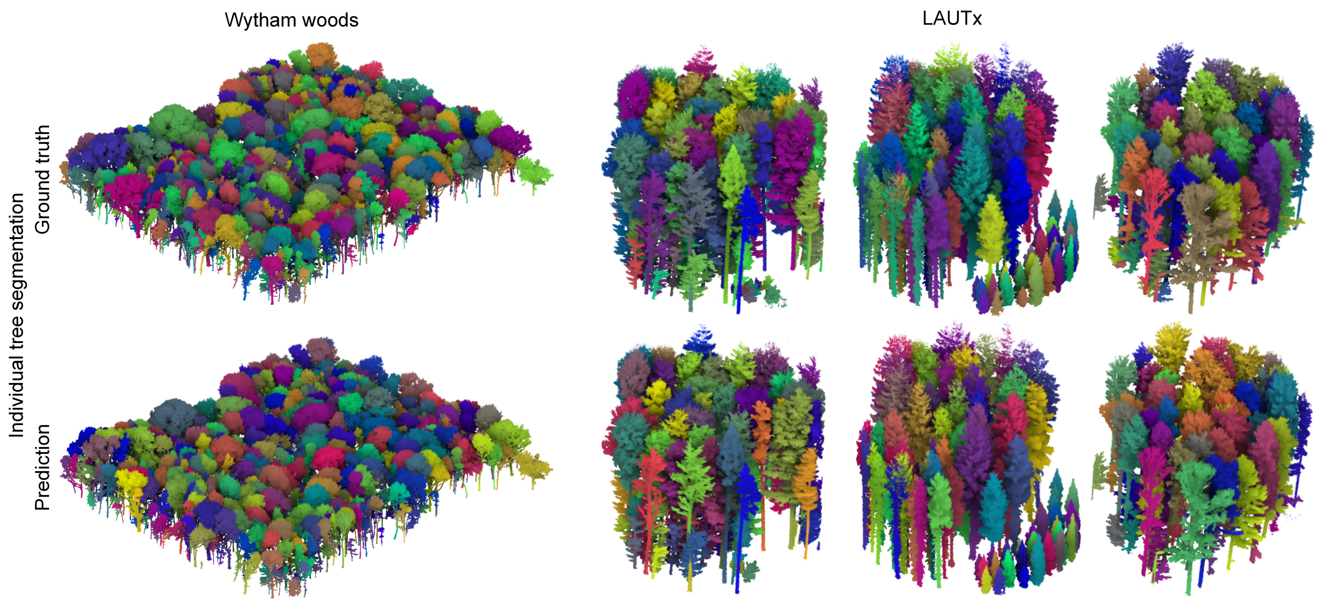

Figure 7. Visualization of ForestFormer3D individual tree segmentation results for Wytham woods and LAUTx. The colors for the individual trees are assigned randomly.

FOR-instance: a UAV laser scanning benchmark dataset for semantic and instance segmentation of individual trees. *arXiv preprint arXiv:2309.01279*, 2023. 2, 3

[7] Adrian Straker, Stefano Puliti, Johannes Breidenbach, Christoph Kleinn, Grant Pearse, Rasmus Astrup, and Paul Magdon. Instance segmentation of individual tree crowns with YOLOv5: A comparison of approaches using the ForInstance benchmark LiDAR dataset. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 9:100045, 2023. 2, 4

[8] Jan Trochta, Martin Krůček, Tomáš Vrška, and Kamil Král. 3D Forest: An application for descriptions of three-dimensional forest structures using terrestrial LiDAR. *PloS one*, 12(5):e0176871, 2017. 2

[9] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *CVPR*, pages 4091–4100, 2019. 1

[10] Maciej Wielgosz, Stefano Puliti, Binbin Xiang, Konrad Schindler, and Rasmus Astrup. SegmentAnyTree: A sensor and platform agnostic deep learning model for tree segmentation using laser scanning data. *Remote Sensing of Environment*, 313:114367, 2024. 2, 3, 4, 5

[11] Binbin Xiang, Torben Peters, Theodora Kontogianni, Frawa Vetterli, Stefano Puliti, Rasmus Astrup, and Konrad Schindler. Towards accurate instance segmentation in large-scale LiDAR point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-1/W1-2023:605–612, 2023. 1

[12] Binbin Xiang, Maciej Wielgosz, Theodora Kontogianni, Torben Peters, Stefano Puliti, Rasmus Astrup, and Konrad Schindler. Automated forest inventory: Analysis of high-density airborne LiDAR point clouds with 3D deep learning. *Remote Sensing of Environment*, 305:114078, 2024. 2, 4, 5

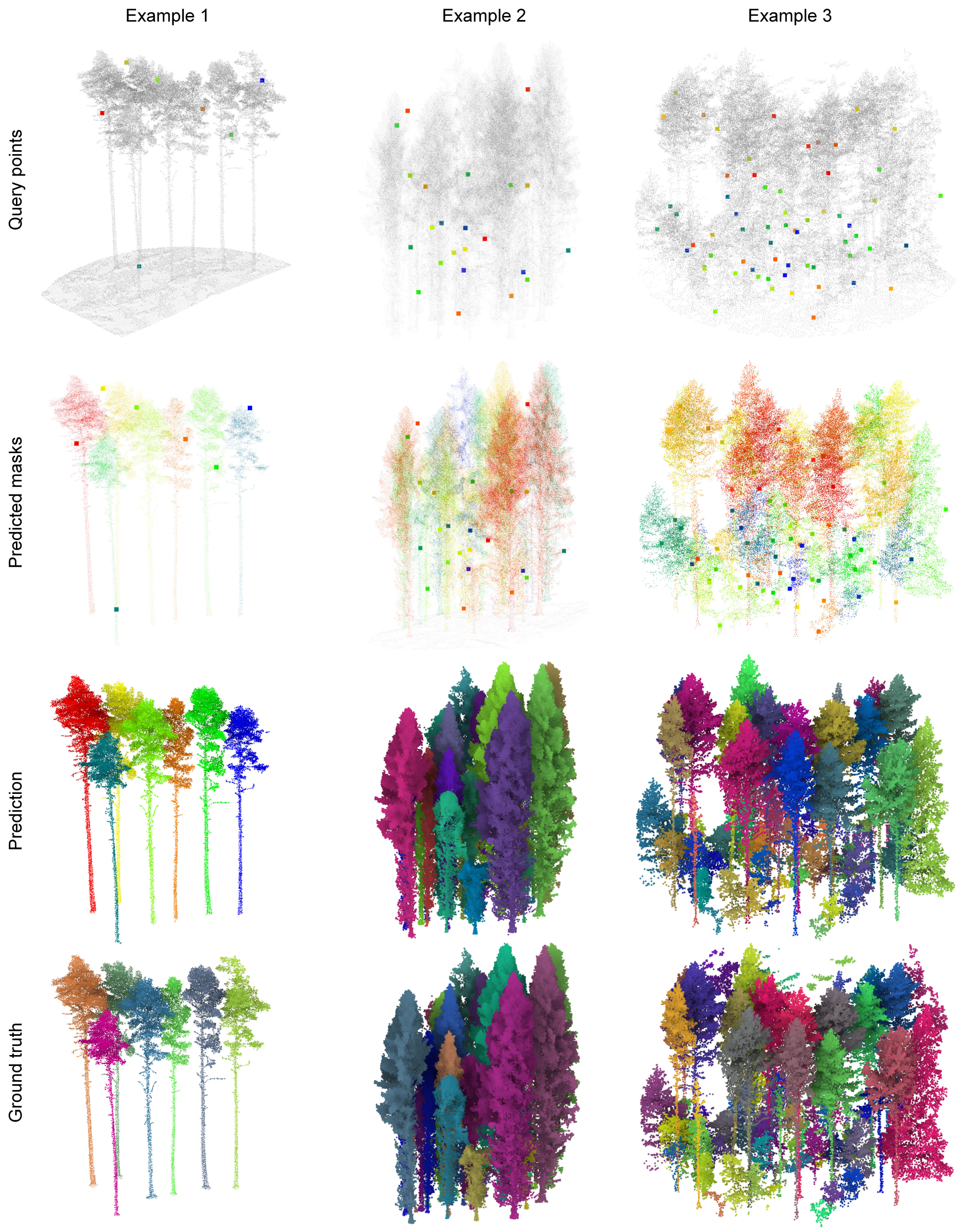

Figure 8. Visual comparison of individual tree segmentation for three representative forest scenes, with each row illustrating a different stage in our segmentation framework. In the first row, the selected ISA-guided query points are highlighted as enlarged, randomly colored points, effectively covering nearly every tree in the scene. The second row shows the predicted masks generated based on the query points from the first row, with each mask color corresponding to its associated query point color. The third row increases color contrast to distinguish each predicted individual tree, with colors again randomly assigned. The fourth row presents the ground truth segmentation with randomly assigned colors to aid comparison.

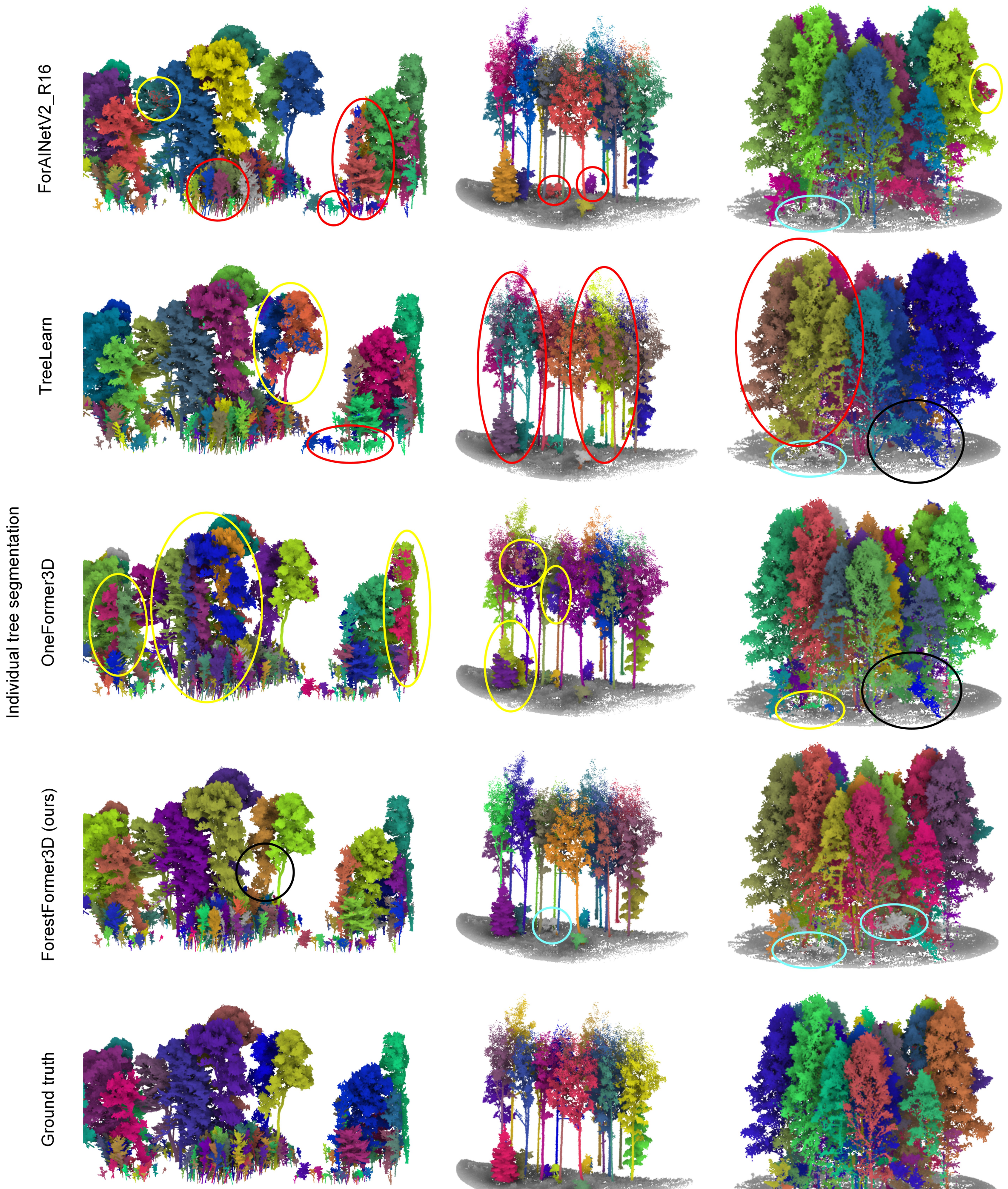

Figure 9. Visual comparison of individual tree segmentation results among ForestFormer3D and baseline methods, with randomly assigned colors representing different tree instances. Common segmentation errors are highlighted using four distinct markers: red circles for under-segmentation, yellow circles for over-segmentation, bright blue circles for undetected trees, and black circles for other errors, such as points from one tree being assigned to a neighboring tree.